9/442,909

```
Set     Items   Description
S1    2892717   STATE? OR STATUS OR CONDITION?
S2    1911002   SEQUENCE? OR ITERATIVE? OR REITERAT? OR NEXT? OR FOLLOWING
            OR ANOTHER
S3    2903057   STOP OR ABORT OR END OR STOPS OR EXIT OR EXITS OR STOPS OR
            TERMINATE?
S4    1453911   NODE? OR APPLICATION? OR EXECUTIBLE OR PROGRAM? OR FILE?
S5       7015   S1 AND S2 AND S3 AND S4
S6      22038   S4(3N)S1
S7        673   S5 AND S6
S8    1551219   COMPUTER? OR PROCESSOR? OR PC OR MICROCOMPUTER? OR NODE? OR
             WORKSTATION? OR WORK()STATION?
S9        340   S7 AND S8
S10     61513   S1(3N)(CHECK? OR VERIF? OR DETERMIN? OR TEST?)
S11        25   S9 AND S10
S12        20   S11 NOT AD=19991118:20021118
S13        20   S12 NOT AD=20021118:20050111
File 347:JAPIO Nov 1976-2004/Aug(Updated 041203)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200501
         (c) 2005  Thomson Derwent
```

13/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

03872260 **Image available**
PROCESSING RESULT INFORMING METHOD FOR UNMANNED SYSTEM

PUB. NO.: 04-237360 [JP 4237360 A]
PUBLISHED: August 25, 1992 (19920825)
INVENTOR(s): HAMANO YUTAKA
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 03-005545 [JP 915545]
FILED: January 22, 1991 (19910122)
INTL CLASS: [5] G06F-015/00; G06F-001/00
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- **Computer Applications**); 45.9 (INFORMATION PROCESSING -- Other
JOURNAL: Section: P, Section No. 1464, Vol. 17, No. 4, Pg. 97, January 06, 1993 (19930106)

ABSTRACT

PURPOSE: To enable an **end** user to **check** the **end state** of a post processing **program** on the preceding date by an **end** user terminal by collectively managing the **end state** of post processing in the preceding date in a processed result management **file** in each subsystem and displaying the **end state** by an **end state** display **program** at the time of connecting respective subsystem terminals on the **next** day.

CONSTITUTION: This unmanned system processed result informing system includes a **status** management **program** 2B for recording the processing **end state** of a batch **program** in each subsystem and a processed result management **file** 4 for storing the record of the processing **end state** , and at the time of connecting an on-line subsystem to an **end** user terminal 5, the **end state** of post processing extracted from the processed result management **file** displayed on the **end** user terminal by the **end state** display **program** .

13/5/10 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012988050 **Image available**
WPI Acc No: 2000-159903/200014
XRPX Acc No: N00-119302

Graphical call stack information generating system for parallel processing system

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: SISTARE S J
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6014514	A	20000111	US 95438437	A	19950515	200014	B

Priority Applications (No Type Date): US 95438437 A 19950515
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6014514	A		19	G06F-011/32	

Abstract (Basic): US 6014514 A

NOVELTY - The control **processor** (11) determines the consolidated **program** routine **sequence** information based on the call stack defining **program** routine **sequence** information retrieved from the processing **nodes** (13). The graph with interconnected graphical **nodes** is generated to represent consolidated **program** routine **sequence** .

DETAILED DESCRIPTION - The processing array comprises multiple processing elements that processes **program** comprising of main routines and subroutines. **Status condition determination** element **determines** the change in processing **status** of each processing **node** and updates the consolidated **program** routine **sequence** based on the change. The processing **node** enters into a **stop condition** , when they execute the **program** which includes **stop** instruction, and **status condition determination** element generates the consolidated **program** routine **sequence** information for the processing **node** which is in **stop condition** . An INDEPENDENT CLAIM is also included for graphical call stack information generating method.

USE - For generating graphical call stack information for parallel processing system.

ADVANTAGE - The debugging arrangement facilitates the debugging of **computer programs** and assist in developing **computer programs** for use in connection with **computer** systems.

DESCRIPTION OF DRAWING(S) - The figure shows the general functional block diagram of parallel processing system.

Control **processor** (11)
Processing **node** (13)
pp; 19 DwgNo 1/6

Title Terms: GRAPHICAL; CALL; STACK; INFORMATION; GENERATE; SYSTEM; PARALLEL; PROCESS; SYSTEM
Derwent Class: T01
International Patent Class (Main): G06F-011/32
File Segment: EPI

13/5/13 (Item 6 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012010496 **Image available**
WPI Acc No: 1998-427406/199836
XRPX Acc No: N98-333632
Digital computer **operating network for interactive debugging of** computer program **- involves beginning searching of main log for each specified registers and memory locations corresponding to time set previously, until value is found or** end **of main log is reached**
Patent Assignee: DIGITAL EQUIP CORP (DIGI)
Inventor: BISHOP J E; CARIGNAN D A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5784552	A	19980721	US 9398501	A	19930728	199836	B

Priority Applications (No Type Date): US 9398501 A 19930728
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5784552	A		46	G06F-011/00	

Abstract (Basic): US 5784552 A

The method involves executing an instruction **sequence** in forward direction, for **determining** current **state** of memory and registers. Each executed instruction is recorded as a main log pre-existing values of any registers and memory locations that are varied by the instruction, where the main log does not include the current **state** . The contents of specific registers and memory locations are reconstructed accordingly. The main log for entries including values are specific registers and memory locations, is then searched.

The searching of main log is started at a location corresponding to specific time set previously and continued until a value is found for the specific register and memory locations or until the **end** of the main log is loaded. When the **end** is reached, a value is obtained from current **state** by each specified registers and memory locations whose value is not found in the main log.

ADVANTAGE - Permits multiple alternative log thereby enabling **pro**grammer to revisit any previously visited location, quickly. Resumes forward execution of **computer program** from updated **state** easily.

Dwg.1/31

Title Terms: DIGITAL; **COMPUTER** ; OPERATE; NETWORK; INTERACT; DEBUG; **COMPUTER** ; **PROGRAM** ; BEGIN; SEARCH; MAIN; LOG; SPECIFIED; REGISTER; MEMORY; LOCATE; CORRESPOND; TIME; SET; VALUE; FOUND; **END** ; MAIN; LOG; REACH
Derwent Class: T01
International Patent Class (Main): G06F-011/00
File Segment: EPI

13/5/19 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX

004574257
WPI Acc No: 1986-077601/198612
XRPX Acc No: N86-056839
Signature analysis for circuit testing - has three registers indicating and applying signature to microprocessor to assess state **of** node
Patent Assignee: STANDARD TEL & CABLES PLC (STTE)
Inventor: BRIDLE P R J; HOVERD T S; PAYNE R D; WADDELL M R R
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2164474	A	19860319	GB 8423264	A	19840914	198612	B
GB 2164474	B	19880413				198815	

Priority Applications (No Type Date): GB 8423264 A 19840914
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
GB 2164474	A		7		

Abstract (Basic): GB 2164474 B

The signature analysis, includes applying a digital data stream from a **node** of a circuit to be tested to a feedback shift register for a preset period or window. At the **end** of this window the register contains a multi-bit number representative of the logic **state** of that **node** . The signature is transferred to **another** register and the test repeated. The signature in the second register is then transferred to a third register and the new signature placed in the second register.

The signatures in the second and third registers are compared by a comparator and when after several signatures have been taken it is indicated that a stable signature has been obtained, that signature is applied to a microprocessor for **checking** to assess the **state** of the **node** under test. The arrangements are included on a chip, and can be on the same chip as a logic circuit to be monitored.

USE - Digital telephone exchange. (7pp Dwg.No.0/4
Title Terms: SIGNATURE; ANALYSE; CIRCUIT; TEST; THREE; REGISTER; INDICATE; APPLY; SIGNATURE; MICROPROCESSOR; ASSESS; **STATE** ; **NODE**
Index Terms/Additional Words: **DIGITAL** ; **TELEPHONE**
Derwent Class: T01; W01
International Patent Class (Additional): G06F-011/30
File Segment: EPI

```
S1      2892717   STATE? OR STATUS OR CONDITION?
S2      1911002   SEQUENCE? OR ITERATIVE? OR REITERAT? OR NEXT? OR FOLLOWING
             OR ANOTHER
S3      2903057   STOP OR ABORT OR END OR STOPS OR EXIT OR EXITS OR STOPS OR
             TERMINATE?
S4      1453911   NODE? OR APPLICATION? OR EXECUTIBLE OR PROGRAM? OR FILE?
S5         7015   S1 AND S2 AND S3 AND S4
S6        22038   S4(3N)S1
S7          673   S5 AND S6
S8      1551219   COMPUTER? OR PROCESSOR? OR PC OR MICROCOMPUTER? OR NODE? OR
              WORKSTATION? OR WORK()STATION?
S9          340   S7 AND S8
S10       61513   S1(3N)(CHECK? OR VERIF? OR DETERMIN? OR TEST?)
S11          25   S9 AND S10
S12          20   S11 NOT AD=19991118:20021118
S13          20   S12 NOT AD=20021118:20050111
S14        9606   S4(2N)(STATE? OR STATUS)
S15         177   S14 AND S9
S16          52   S15 AND (CHECK? OR VERIF? OR DETERMIN? OR TEST OR TESTS OR
               TESTING OR TESTED)
S17          41   S16 AND IC=G06F?
S18          34   S17 NOT AD=19991118:20021118
S19          33   S18 NOT AD=20021118:20040122
S20          26   S19 NOT S11
S21          26   IDPAT (sorted in duplicate/non-duplicate order)
S22          26   IDPAT (primary/non-duplicate records only)
File 347:JAPIO Nov 1976-2004/Aug(Updated 041203)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200501
         (c) 2005  Thomson Derwent
```

22/5/16 (Item 16 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

03157028 **Image available**
DUPLEX **PROCESSOR**

PUB. NO.: 02-132528 [JP 2132528 A]
PUBLISHED: May 22, 1990 (19900522)
INVENTOR(s): HAYASHI TAKAO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 63-285791 [JP 88285791]
FILED: November 14, 1988 (19881114)
INTL CLASS: [5] **G06F-011/16** ; **G06F-015/16** ; **G06F-015/16**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units); 45.4 (INFORMATION PROCESSING -- **Computer Applications**)
JOURNAL: Section: P, Section No. 1088, Vol. 14, No. 360, Pg. 69, August 03, 1990 (19900803)

ABSTRACT

PURPOSE: To improve the system reliability by securing such a constitution where an active system **processor checks** the normalcy of a **test program** which is carried out by a stand-by system **processor** based on the executing address of the **test program** .

CONSTITUTION: A stand-by **processor** 1-2 contains a trouble detecting **program** stored in a memory 2-2. This **program checks** whether a stand-by system 1-2 is kept in a **program** run **state** or not. If so, the system 1-2 is stopped. The address set when the system 1-2 is stopped is informed to an active **processor** 1-1 via a both-system communication line 6. The **processor** 1-1 **checks** the **stop** address based on the memory allocating information on the trouble detecting **program** . If the **stop** address is normal, both an on-line **program** and the trouble detecting **program** are carried on. While a **program** is started for diagnosis of the **processor** 1-2 when the **stop** address is not normal. Thus it is possible to carry out the trouble detecting function of the **processor** 1-2 with higher safety and to improve the system reliability.

22/5/19 (Item 19 from file: 347)
DIALOG(R)File 347:JAPIO

02699245 **Image available**
SYSTEM FOR CONTROLLING RE-START OF DATA **PROCESSOR**

PUB. NO.: 63-316145 [JP 63316145 A]
PUBLISHED: December 23, 1988 (19881223)
INVENTOR(s): MIYAZAKI MASAMITSU
APPLICANT(s): HITACHI DENSHI LTD [000542] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-151197 [JP 87151197]
FILED: June 19, 1987 (19870619)
INTL CLASS: [4] **G06F-011/30** ; **G06F-011/14**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units)
JOURNAL: Section: P, Section No. 857, Vol. 13, No. 157, Pg. 118, April 17, 1989 (19890417)

ABSTRACT

PURPOSE: To improve the automatic recovery capacity of a device for the runaway of a **program** , by monitoring the abnormal execution **state** of the **program** by both the **program** of high hierarchy and the **program** of low hierarchy.

CONSTITUTION: When the output of a reset signal 1.2 from a data **processor** 1 **stops** , a monitoring timer 3 times up, and outputs a reset signal 3.1. The signal 3.1 resets the **processor** 1. Also, the **processor** 1 outputs a count up signal 1.3 from the **program** of low hierarchy to a counter 4. Meanwhile, a timer 6 **checks** a timer value at every inputting of the signal 1.2 and when the timer 6 times up, the timer outputs a qualification start up signal 6.1 to a register 5. The register 5 inputs the value 4.1 of the counter 4 by the signal 6.1, and compares it with a preceding value inputted by the signal 6.1 just before. When they coincide with the result of comparison, it decides that the execution of the **program** of low hierarchy is abnormal, and outputs a re-start signal 5.1 to the **processor** 1. In such a way, it is possible to combine duplicated monitoring re-start functions.

22/5/20 (Item 20 from file: 347)
DIALOG(R)File 347:JAPIO

02648437 **Image available**
FAULT DETECTING CIRCUIT FOR **PROCESSOR** DEVICE

PUB. NO.: 63-265337 [JP 63265337 A]
PUBLISHED: November 01, 1988 (19881101)
INVENTOR(s): MIYAHARA YOSHIRO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-100554 [JP 87100554]
FILED: April 22, 1987 (19870422)
INTL CLASS: [4] **G06F-011/00** ; **G06F-009/06**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units)
JAPIO KEYWORD:R131 (INFORMATION PROCESSING -- **Microcomputers** & Microprocessers)
JOURNAL: Section: P, Section No. 833, Vol. 13, No. 82, Pg. 120, February 23, 1989 (19890223)

ABSTRACT

PURPOSE: To detect a fault caused by a branch error at an early **state** and with high probability by writing previously the information referring to the flow of a **program** into a **program** memory and collating this stored information with a counter control signal when the **program** is executed.

CONSTITUTION: The information showing whether or not a time point when the contents of a **program** address are read out in a **program** executing **state** for each **program** address must be coincident with a time point right after the contents of the **program** address preceding by a step, i.e., a **program check** bit signal 12 for a **program** flow is previously stored in a **program** memory 1. When a **program** is carried out, the signal 12 serving as the information on the **program** is read out simultaneously with the **program** itself. Then a collating circuit 5 collates the signal 12 with a 1-cycle preceding counter control signal 13, i.e., the information showing whether a **program** address counter 2 is advanced by one or not right before the signal 12 is read out. Thus a branch error can be detected while a **program** is carried out. Furthermore the deciding output of the circuit 5 serves as a fault detecting signal 14 and is used to forcibly **stop** the execution of a **program** , etc.

22/5/23 (Item 23 from file: 347)
DIALOG(R)File 347:JAPIO

02370445 **Image available**
MEMORY ACCESS **STATE** MONITORING DEVICE

PUB. NO.: 62-287345 [JP 62287345 A]
PUBLISHED: December 14, 1987 (19871214)
INVENTOR(s): HATTORI SHIGERU
APPLICANT(s): FUJI ELECTRIC CO LTD [000523] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 61-131503 [JP 86131503]
FILED: June 06, 1986 (19860606)
INTL CLASS: [4] **G06F-011/30**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- **Microcomputers** & Microprocessers)
JOURNAL: Section: P, Section No. 708, Vol. 12, No. 178, Pg. 26, May 26, 1988 (19880526)

ABSTRACT

PURPOSE: To easily understand a loop **state** of a **program** to be **tested**, and also, to curtail a trouble for **program** development, by starting a monitoring operation, when the number of times of coincidence of an address by which a memory to be monitored is brought to an access and the first address which is set in advance has reached the first number of times of coincidence, and ending the monitoring operation, when the number of times of coincidence of the address by which the memory to be monitored is brought to an access and the second address which is set in advance has reached the second number of times of coincidence.

CONSTITUTION: Each means 30-36 generates a buffer on-signal 45, and a buffer off- signal 46 in accordance with various **conditions** and executes on an off of buffers 8A, 9A, namely, a start and **stop** control of a memory monitoring operation. Among said means, the means 30 and 31 are a start address coincidence circuit and a **stop** address coincidence circuit, respectively, and a start address and a **stop** address are set, respectively, through a **microcomputer** 2, and whenever the value of an address signal on an address bus 24 coincides with this set address, an address coincidence signal 30a and 31a are supplied to a start address coincidence counter 33 and a **stop** address coincidence counter 35 of the **next** stage, respectively.

22/5/24 (Item 24 from file: 347)
DIALOG(R)File 347:JAPIO

02103139 **Image available**
STOP OPERATION SUPPRESSING CONTROL SYSTEM

PUB. NO.: 62-020039 [JP 62020039 A]
PUBLISHED: January 28, 1987 (19870128)
INVENTOR(s): SAKAI TAKASHI
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 60-159819 [JP 85159819]
FILED: July 19, 1985 (19850719)
INTL CLASS: [4] **G06F-011/28**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units)
JOURNAL: Section: P, Section No. 589, Vol. 11, No. 197, Pg. 85, June 25, 1987 (19870625)

ABSTRACT

PURPOSE: To make the operation easy to improve the **test** efficiency by making a **stop** command ineffective in a prescribed **program** execution **state** if the **stop** command for **test** is set.

CONSTITUTION: A **status** bit 21 is one bit of a **program status** word and is turned on in the problem **state** and is turned off in the supervisor **state** . When the **stop** operation is designated from a **testing** equipment 1 by an instruction **stop** designating line which instructs a **processor** 29 to **stop** execution for every instruction or an address **stop** designating line 4 which instructs the **processor** 20 to **stop** execution for coincidence of access address, a **stop** designation latch 5 is set. A gate 22 operates AND among the signal of the **stop** designation latch 5, the signal on an instruction **end** line 7 which is outputted from an execution control part 6 and indicates the execution **end** of every instruction, and the **status** bit 21 and outputs the result to a signal line 23. A **stop** control part 10 is started by the turn-on signal on the signal line 23 to **stop** the advance of the execution control part 6.

22/5/2 **(Item 2 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

010733267 **Image available**
WPI Acc No: 1996-230222/199623
XRPX Acc No: N96-193348
Automatic computer **software** testing **system for software product development - uses** deterministic **acceptance** test **and random command** sequence **selections with results analyser** checking test **parameters** following **execution of predetermined and random** test **commands**
Patent Assignee: MICROSOFT CORP (MICR-N)
Inventor: CROSS N; TIERNEY J R
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5513315	A	19960430	US 92995746	A	19921222	199623	B

Priority Applications (No Type Date): US 92995746 A 19921222
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5513315	A		17	G06F-011/00	

Abstract (Basic): US 5513315 A

The system includes a preliminary **test** file which contains a series of predetermined **computer** software commands for **testing** initial **computer** software parameters. An operational profile containing a list of **end** -user choices is generated in response to a particular **program state** of the **computer** software. The list is constructed according to probabilities of **end** -user choices in response to the particular **program state** . A series of commands are randomly selected from the operational profile for the **computer** software to execute and are recorded by a tracker log **file** . A portion of the tracker log **file** is selected.

The selection of the predetermined commands, the random series of commands, and at least the selected portion of the tracker log are simulated. At least one **test** parameter is analysed after execution of the simulation to **determine** if each of the predetermined commands was properly executed by the **computer** software. A **test** result is generated and a **test** log **file** recording is made. An error recovery unit automatically restarts the **computer** software to continue **testing** the **computer** software if one of the random series of commands was not properly executed by the **computer** software.

ADVANTAGE - **Applications programs** can be **tested** on multiple machines to decrease overall **testing** time. Low cost.

Dwg.2/4
Title Terms: AUTOMATIC; **COMPUTER** ; SOFTWARE; **TEST** ; SYSTEM; SOFTWARE; PRODUCT; DEVELOP; ACCEPT; **TEST** ; RANDOM; COMMAND; **SEQUENCE** ; SELECT; RESULT; ANALYSE; **CHECK** ; **TEST** ; PARAMETER; FOLLOW; EXECUTE; PREDETERMINED; RANDOM; **TEST** ; COMMAND
Derwent Class: T01
International Patent Class (Main): **G06F-011/00**
File Segment: EPI

05279634 **Image available**

COMPUTER STARTING-UP METHOD OF MULTICOMPUTER SYSTEM

PUB. NO.: 08-235134 [JP 8235134 A]
PUBLISHED: September 13, 1996 (19960913)
INVENTOR(s): TSUKAHARA HIROTO
KANEKO SHIGENORI
MIYAO TAKESHI
KATO SUNAO
KASASHIMA HIROKAZU
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
HITACHI PROCESS COMPUT ENG INC [485525] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 07-036485 [JP 9536485]
FILED: February 24, 1995 (19950224)
INTL CLASS: [6] **G06F-015/16** ; **G06F-001/00** ; **G06F-011/20**
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- **Computer Applications**);
45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units);
45.9 (INFORMATION PROCESSING -- Other

ABSTRACT

PURPOSE: To suppress influence on other **computers** if a fault is generated right after a transaction **program** is started to repeat resetting and restarting and lighten the load on the whole system by stopping reporting the survival of a **computer** where a fault occurs, and then disconnecting this **computer** from mutual life/death monitoring and evading the start of the transaction **program** .

CONSTITUTION: When a **computer** 1 (101) is powered ON and a multiple system operation **state** managing **program** 1 starts the starting-up process of its own system 102, a data **check** 103, whether or not there is startup in-process data 113 being start data in an auxiliary storage device 110, is made first. When the data is present, it is judged that the **computer** 1 (101) **stops** during the start-up process, and the startup of its own system is ended 114. When the startup in-process data 113 is not present, on the other hand, startup in-process data 113 is newly written 104 in the auxiliary storage device 110, and a survival information **program** 105 and a life/dead monitoring **program** 106 are started to start the transaction **program** .

22/5/14 (Item 14 from file: 347)
DIALOG(R)File 347:JAPIO

03450747 **Image available**
COMPUTER SYSTEM AND TRACE DEVICE

PUB. NO.: 03-113647 [JP 3113647 A]
PUBLISHED: May 15, 1991 (19910515)
INVENTOR(s): AOKI KATSUICHI
WATABE SHINYA
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
HITACHI MICRO COMPUT ENG LTD [470864] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 01-252641 [JP 89252641]
FILED: September 28, 1989 (19890928)
INTL CLASS: [5] **G06F-011/28**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic **Sequence** Units)
JOURNAL: Section: P, Section No. 1237, Vol. 15, No. 316, Pg. 75, August 13, 1991 (19910813)

ABSTRACT

PURPOSE: To attain trace that can recognize a time relation between data which respective **processors** accumulate by providing trace means accumlating hysteresis information of self processings and common time signals for trace system as traced signals and means independently controlling the interruption or execution of the trace means for respective **processors** .

CONSTITUTION: The tracer control parts 103-1, 103-N and 103-S trace a system tracer control part and **stop** -control all the tracers 104-1, 104-N and 104-S when a phenomenon which comes to be a system **check stop** cause occurs in respective **processors** or a storage device 101-S, for example. Then, trace suppression signal lines 106-1, 106-N and 106-S suppress a trace write signal by turning on instruction processing parts 102-1 and 102-N or a storage control part 102-S when a **program state** word wait bit is '1' and the like for example. Thus, trace that can recognize the time relation between data which respective **processors** accumulate efficiently is executed.